

Access Tip

Access 2.0 provides enhancements for both users and application developers

At the end of March, users and developers from across the US assembled at the Microsoft Tech•Ed conference in New Orleans to hone their skills with Microsoft products. They also learned a lot about the new version of Access. Microsoft announced Access 2.0 and showcased the product throughout the conference.

In this article, we'll give *you* a look at the new version. We'll show you many of the user-oriented enhancements. We'll also discuss what this new version means to developers. However, please keep in mind that we prepared this report for you with pre-release software. Access' developers may have changed some aspects of the program in the final release.

For the user

When you launch Access 2.0 for the first time, you'll notice subtle but significant changes in the interface. We'll start our overview with a description of these enhancements and how they will help you do your work.

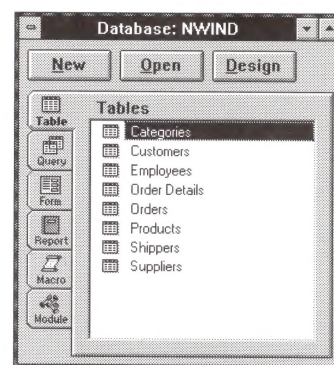
Access 2.0 modified interface

The first change you'll notice is in the Database window. It now looks like a notebook with tabs marking each type of object. Figure A shows NWIND.MDB's Database window while it lists tables. The Table button is connected to the table list like a Post-itTM stuck on a page in a spiral notebook. All the Database window's buttons are connected to their lists of objects in this way.

IN THIS ISSUE

- Access 2.0 provides enhancements for both users and application developers 1
- Converting your Access 1.x database files to Version 2.0 7
- Removing a text box from the tab order with the Enable and Locked properties 8
- A double-click shortcut for bringing a subform or subreport into Design View 9
- Printing forms with white backgrounds 10
- Printing addresses directly onto envelopes 12
- Defining a combo box's selection list based on another control in the form 13
- The ANALYZER.MDA utility lets you print out table specs and more 14

Figure A



The Database window has a new look in Access 2.0.

The tool bars have also changed. As you know, a tool bar is a row of buttons just below the menu bar that provides fast access to relevant operations. Well, tool

bars still provide fast access to the commands. However, they don't necessarily reside at the top of the screen in Version 2.0. You can drag a tool bar out of its position and let it float with the other Access objects. For instance, Figure B shows the new format of the Database tool bar.

Figure B



The tool bar has a new format in Access 2.0.

In this state, the tool bar works just as the tool box does while you're designing a form or report. In fact, the tool box is now just a tool bar that appears in a window by default. You can put it back up with the form- or report-design tool bar if you want.

Another change to the interface is the addition of shortcut menus. When you right-click an object, a shortcut menu appears under your mouse pointer and provides common operations you could perform on that object. For example, when you right-click an object in the Database window, a shortcut menu appears that offers Rename and Print commands, among others. Finally, Access 2.0 displays a *tool tips* message when you let the mouse rest on a tool bar button. Tool tips are messages that display the button's name and disappear as soon as you move the mouse.

New wizards help design tables, queries, and even controls

Let's now move on to the major enhancements. One of the most important changes

in Access 2.0 is the substantial number of new wizards. You'll still find the familiar wizards for creating forms and reports—you'll also find table and query wizards. These wizards work just like the form and report wizards: When you create a new object, Access will display a dialog box that lets you choose whether to use a wizard or start from scratch.

Table wizards let you pattern a new table after a predefined table template. You pick a template from over 25 options and then select from that template the fields you want to include in the final table. Query wizards help you create several types of queries—crosstab queries, select queries that return duplicate records, and more.

Furthermore, Access 2.0 provides four Control wizards, which help you during form and report design. When you create a list box, combo box, option group, or command button, a control wizard will appear and ask you to supply the information necessary to define the control. It then creates the control. If you want to define controls from scratch, you can turn off this feature by clicking a button in the tool box.

Map your tables' relationships with a graphical relationships window

Access 2.0 also improves how you define relationships among tables. As you may

Inside MICROSOFT ACCESS™

Inside Microsoft Access (ISSN 1067-8204) is published monthly by The Cobb Group.

Prices Domestic \$59/yr. (\$7.00 each)
Outside US \$79/yr. (\$8.50 each)

Phone Toll free (800) 223-8720
Local (502) 491-1900
Customer Relations Fax (502) 491-8050
Editorial Department Fax (502) 491-4200

Address You may address tips, special requests, and other correspondence to
The Editor, *Inside Microsoft Access*
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

For subscriptions, fulfillment questions, and requests for bulk orders,
address your letters to
Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

Advertising For information about advertising in Cobb Group journals, contact
Tracee Bell Troutt at (800) 223-8720, extension 430.

Postmaster Second class postage is pending in Louisville, KY. Send address changes to
Inside Microsoft Access
P.O. Box 35160
Louisville, KY 40232

Copyright

Authorized Canada Post International Publications Mail (Canadian Distribution)
Sales Agreement #XXXXXX CANADA GST #123669673. Send returns to
Canadian Direct Mailing Sys. Ltd., 920 Mercer Street, Windsor, Ontario, N9A
7C2. Printed in the USA.

Copyright © 1994, The Cobb Group. All rights reserved. *Inside Microsoft Access* is an independently produced publication of The Cobb Group. The Cobb Group reserves the right, with respect to submissions, to revise, republish, and authorize its readers to use the tips submitted for both personal and commercial use.

The Cobb Group, its logo, and the Satisfaction Guaranteed statement and seal are registered trademarks of The Cobb Group. *Inside Microsoft Access* is a trademark of The Cobb Group. Paradox is a registered trademark of Borland International. dBASE III and dBASE III PLUS are registered trademarks of Ashton-Tate, a Borland International company. Microsoft, MS-DOS, and Access are registered trademarks of Microsoft Corporation. Microsoft Windows and Word for Windows are trademarks of Microsoft Corporation.

Staff

Editor-in-Chief David Brown
Editors Meredith Little
Polly Blakemore
Elizabeth Welch
Maureen Spencer
Marguerite Stith
Production Artists Karl Feige
Design Elayne Noltemeyer
Managing Editor Marjorie Glassman
Circulation Manager Jeff Yocom
Editorial Director Mark Crane
Publishers Jon Pyles

Back Issues

To order back issues, call Customer Relations at (800) 223-8720. Back issues cost \$7 each, \$8.50 outside the US. You can pay with MasterCard, VISA, Discover, or American Express, or we can bill you. Please identify the issue you want by the month and year it was published. Customer Relations can also provide you with an issue-by-issue listing of all articles that have appeared in *Inside Microsoft Access*.

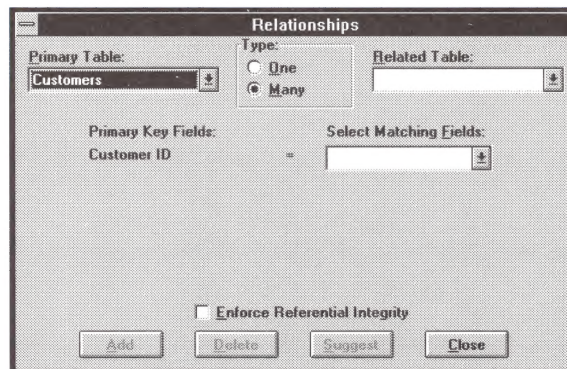
know, you define relationships in versions 1.0 and 1.1 by using the Relationships dialog box, shown in Figure C. (This figure shows information about the relationship between the NWIND.MDB database's Customers and Orders tables.) When you select a table, that table's key field appears under the Primary Key Fields label. You then select a detail table and use the Select Matching Fields combo box to select the field that matches the main table's primary key field.

In Version 2.0, you use the Relationships window shown in Figure D. The graphical presentation provides information about all the database's relationships at a glance. Note the symbols at either end of the join lines: They tell you whether the relationship is one-to-one, one-to-many, or many-to-many. Join lines that have a 1 at each end represent one-to-one relationships. If you see an infinity symbol at one end of the join line, the join lines represent a one-to-many relationship. If you see two infinity symbols (one at each end), the tables have a many-to-many relationship. As you can see, the NWIND.MDB database defines only one-to-many relationships.

To see the details of a particular relationship, you double-click the join line. For instance, when you double-click the join line between the Customers and Orders tables, the window shown in Figure E appears. The Table/Query and Related Table/Query list boxes show you the relationship's linking fields. The window also includes options that let you define various aspects of the relationship, such as referential integrity and cascade delete and cascade update options.

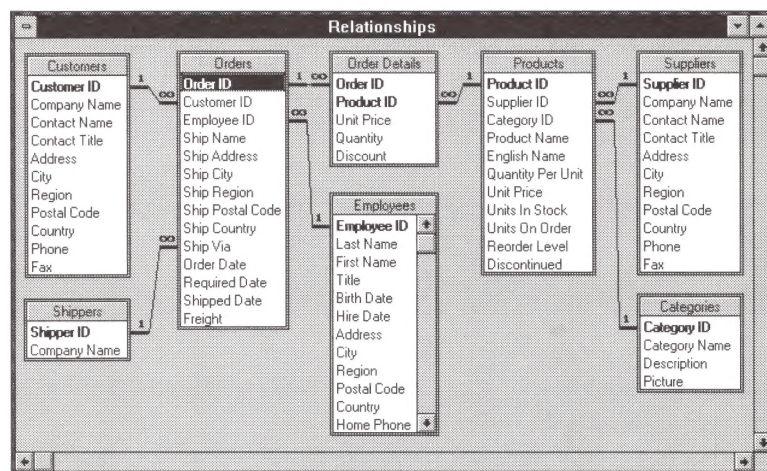
You can also set the type of join that links the tables by clicking the Join Type... button. Clicking the button displays the Join Properties dialog box, which you may be familiar with if you've ever defined a join type in a query. Figure F shows the Join Properties dialog box to define the relationship's join type.

Figure C



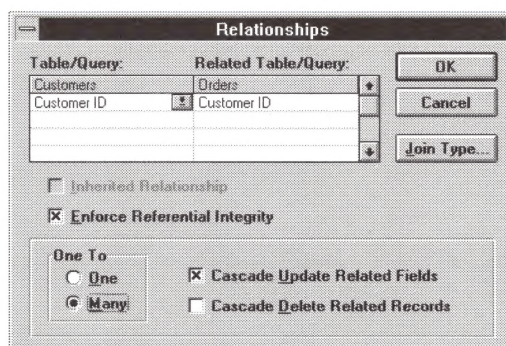
In versions 1.0 and 1.1, the Relationships dialog box lets you match linking fields by using combo boxes.

Figure D



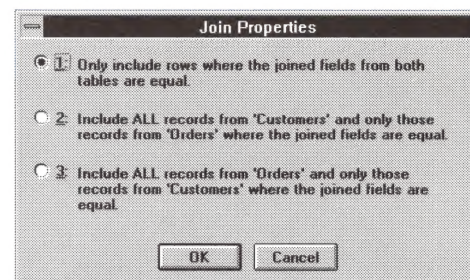
The new Relationships window shows you all the database's relationships at a glance.

Figure E



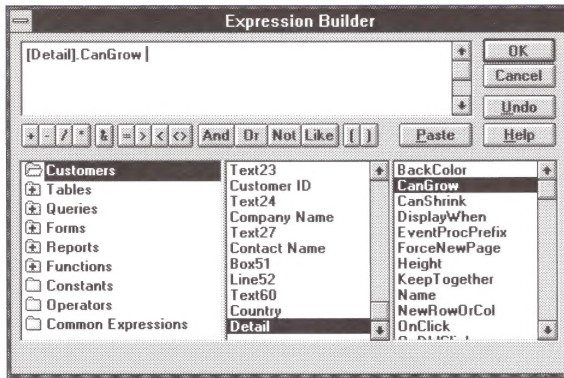
This window, which lets you customize a relationship, appears when you double-click the join line.

Figure F



You use the familiar Join Properties dialog box to define the relationship's join type.

Figure G



The Expression Builder lets you look up components of the database objects.

box for the Customers-Orders relationship in the NWIND.MDB database.

The new Expression Builder provides a reference to object and control names

We'll now turn to a new feature that can dramatically simplify form and report design. If you often use expressions in your forms and reports—and in the macros you create to manage events—you'll appreciate the new Expression Builder. It provides a facility for looking up objects, controls, and properties while you create your expression.

How does the Expression Builder simplify design? No more typos! If you're like most

users, you've mistyped a long identifier in an expression and spent way too much time tracking down your error. If you double-click a control in the Expression Builder, the control's identifiers will automatically appear in your expression.

Figure G shows how the Expression Builder looks while you work in the NWIND.MDB sample database. You type your expression in the text box at the top and look up objects and controls in the

list boxes at the bottom. The leftmost column lists the objects in the database. The middle and right columns provide the components of those objects you can reference in expressions. For example, if you select the Customers form in the left column, the middle column will list the form controls. When you select a control, the right column will list the properties of that control.

The new Output To... command lets you write objects to RTF and XLS files

Access 2.0 also provides new output capabilities. In addition to sending forms, reports, and datasheets to the printer, you can output data to an RTF or XLS file. You right-click the object you want to output and choose the Output To... option. After you choose a file format and provide a filename, Access converts the object to the format you selected and creates the file. You can then open the file in Word or Excel to work on the data by using that application's specific features. You can use Word to spruce up report output for an important presentation. You can use the unique spreadsheet capabilities of Excel to analyze the data. You can also use the Output To... command to write data to a simple text file.

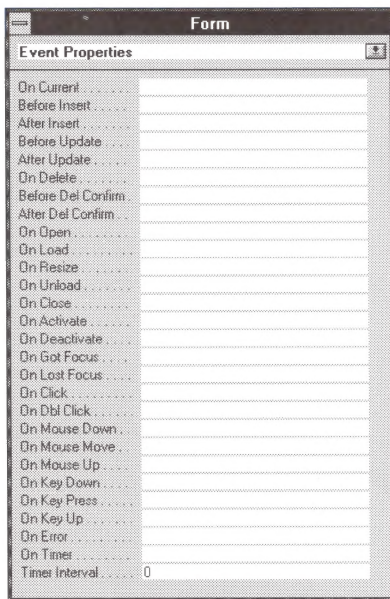
For the developer

If you develop applications in Microsoft Access, you'll enjoy all the new features that Microsoft geared toward the user. However, Microsoft has also added many important enhancements for the developer. We'll next describe several of the most important ones.

New events increase your control on Access forms

Possibly the biggest enhancement is Access' adaptation of the Visual Basic event model. You can control many new events while the user navigates a form. Figure H shows the form's property sheet that lists the form- and record-level events. Notice that you can also trap keystrokes and mouse events. There's even an On Timer event, which you can set to trigger an event after a specified amount of time has elapsed.

Figure H



Access 2.0 adds many new event properties that let you trap form- and record-level events.

Also, note that this version has changed several property names. Table A lists the new names.

Event procedures

Not only has the event model changed, but the way Access stores your Access Basic code has changed as well. In Version 1.x, you create a module in which you develop your functions. After you create the functions, you open your forms in Design View and assign your functions to the properties for which you designed them.

On the other hand, in Access 2.0, you create event procedures. Event procedures are a special type of sub procedure that Access stores directly in the form object rather than in a module. Also, you don't name these procedures: When you create one, Access generates the name in the Sub statement that defines the procedure. However, it doesn't show the procedure name in the property sheet. Access simply displays *[Event Procedure]* as the event property's value. Internally, Access keeps track of which event procedure to call.

Let's look at a quick example. Suppose you want to create an event procedure for the On Click event of a button named Close. You select the button, open the property sheet, click the Builder button (🔧) next to the On Click property, and choose Code Builder in the pop-up menu. Access will display the window shown in Figure I. Notice that the event procedure's name already resides in the Sub statement. When you close the window, the On Click property will contain the value *[Event Procedure]*, as shown in Figure J.

Set almost any form or report property at runtime

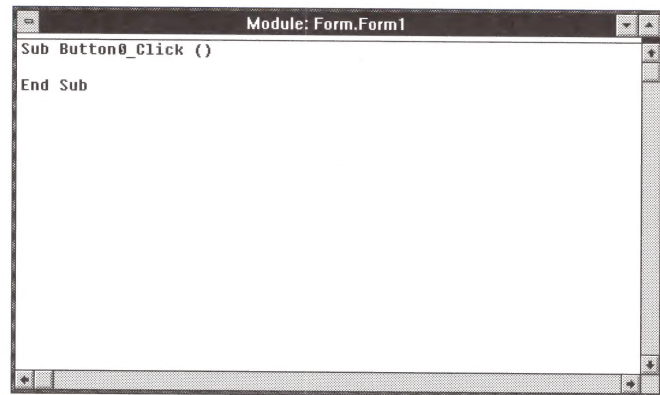
Access 2.0 also addresses one of the most common Access 1.x complaints—that you can set only a couple of form properties at runtime. Setting the Visible property to hide and reveal controls while the user operates on the form is about the most powerful thing you can do with form properties.

However, in Access 2.0, you can set almost any property. For instance, you can even set a form's Record Source property on the fly. Suppose that you routinely transfer data from the primary

Table A

Version 1.x Property Name	Version 2.0 Property Name
Control Name	Name
FormName	Name
OnInsert	BeforeInsert
OnMenu	MenuBar
OnPush	OnClick
Scaling	SizeMode
UpdateMethod	UpdateOptions

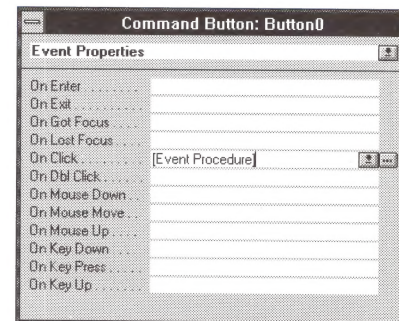
Figure I



Access generates event procedure names automatically.

data table to an archive table. Occasionally, the user will need to look up an old record. In Version 1.x, you might create a button that opens a Customer Archive form. In Version 2.0, you can put an Archive Toggle button on the form. You can create a macro or event procedure for the button that toggles the Record Source property between Customer and Customer Archive. That way, when users need to look up old orders, they can click the button to view the archived data. When they've finished, they can toggle back to current data by clicking the button again.

Figure J



When you create an event procedure, the property sheet shows the label *[Event Procedure]* in the event property.

Data definition queries let your application create tables

Furthermore, Access 2.0 has a couple of new query types that help out the developer. The first type we'll describe is the data definition query. Your application can use these queries to create, redesign, and even delete Access tables on the fly.

When you create a data definition query, Access opens an empty window in which you type the SQL command that creates or modifies the tables. You may use the SQL command CREATE TABLE, CREATE INDEX, ALTER TABLE, or DROP.

The NWIND.MDB database provides an example of a data definition query. Figure K shows a sample query named Data Definition Query that creates a table named Friends.

SQL pass-through capability

The other new query type is the SQL pass-through query, which passes your query to the SQL server for processing. As you may know, if you create an ordinary select query that includes an attached SQL table, Access may download some of the remote data so that it can process the query locally. By using a pass-through query, you tell the SQL server to process the query.

To create a pass-through query, you must supply your query's SQL code. Access provides a blank window in which you type the SQL statement.

The Menu Builder lets you design menus graphically

If you've created many menus in Version 1.x, you'll be happy to see the new Menu Builder that ships with Access 2.0. Access hasn't changed how it stores menus—by using macro groups and AddMenu actions. However, the Menu Builder provides an easy way to organize the structure of the menu. It also generates all the macro groups and the main menu-bar macro for you.

Furthermore, the Menu Builder lets you begin menu design with an Access menu bar. You can delete the Access menu commands you don't want the user to run and substitute your own custom menu commands. For example, Figure L shows the Menu Builder when you design a menu based on the Form menu bar.

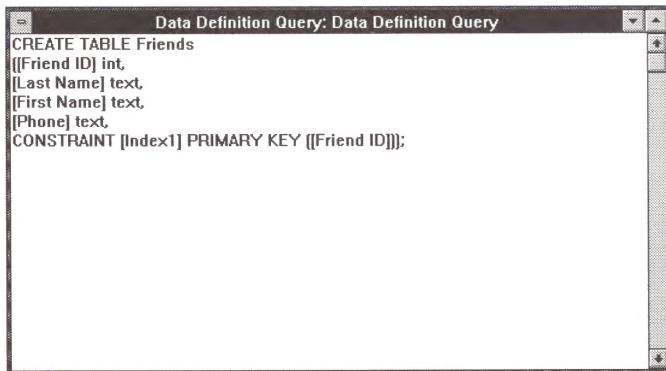
For everyone—performance

Finally, we'll comment on performance—an issue everyone wants to know about. Access 2.0 incorporates FoxPro's Rushmore technology. Rushmore is the query optimizer that put FoxPro at the top of the heap of PC-based databases in terms of query performance. The Rushmore technology will speed up some types of queries dramatically.

Conclusion

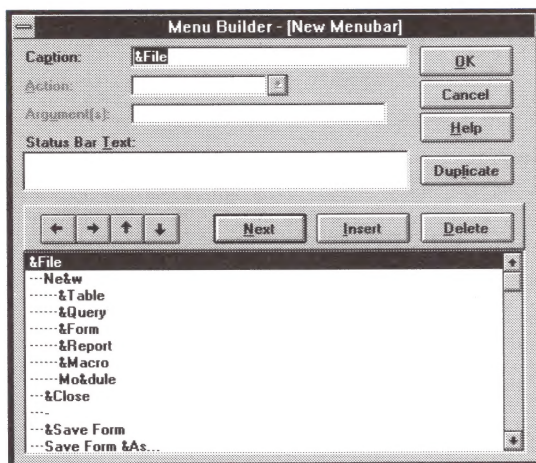
In summary, Access now has a new look and plenty of improvements to the interface that make navigating Access much easier. It also features new wizards and builders that increase productivity when you design the objects in your database. Finally, with all the new development tools, Access 2.0 provides a rich environment for database application development. ❖

Figure K



This data definition query creates a table named Friends.

Figure L



The Menu Builder lets you begin with an Access menu bar and then customize it by deleting the unwanted commands and adding your own commands.

Converting your Access 1.x database files to Version 2.0

With all the enhancements we described in our lead article, you may have assumed that you won't be able to use your Version 1.x database files in Version 2.0. Well, that's half true. You will be able to work with Version 1.x database files. However, you won't be able to enter Design View for any of the objects. In other words, you won't be able to take advantage of any of those new features. In this article, we'll show you how to convert your Version 1.x database files so you can use them in Access 2.0 along with the new features.

Using the Convert Database command

To convert your Version 1.x database files when you upgrade to Access 2.0, you issue the Convert Database command from the File menu *before* you open a database file. (If you open a database first, the File menu won't contain that command.) When you run the Convert Database command, Access displays two dialog boxes in sequence: In the first, you supply the name of the database file you want to convert; in the second, you supply the name of the file in which you want to store the converted database.

Although you can tell Access to overwrite your original database with the Version 2.0 database, you probably shouldn't do so—at least not the first time you convert a database. Access 2.0 is a new program and, as such, it may change the behavior of some database objects. If you store the Access 2.0 version of your database under a new filename, you can then test the upgraded database to ensure it's working properly.

Notes on converting databases

There are a few possible glitches you should consider when converting a database. We'll list them here in no particular order:

- Once you convert a database to the Access 2.0 file format, you can't

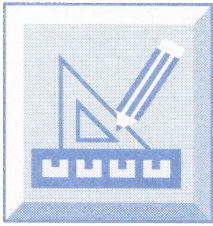
convert it back. Also, you can't open a converted database in either Version 1.0 or 1.1.

- Access 2.0 applies validation rules more rigorously than Access 1.x does. If it finds records that don't meet the conditions of your validation rules, Access records the error in a new table named ConvertErrors.
- If your database accesses data through attached tables, you don't need to convert the databases that hold those tables. On the other hand, once you've converted a database, the Version 1.x database can't attach its tables anymore.
- If you've implemented Access security for your database, only users with Modify Design permissions can convert a database.

Strategies for converting your databases

We'll conclude by providing a couple of strategies for converting your databases. First, you should always make a copy of your database file before converting the file. That way, if anything you don't like happens, you can always return to the Version 1.x database and decide how to proceed. Also, don't add any data to your tables while you're testing the Access 2.0 version of the database file. Your copy of the Version 1.x database won't receive the new data.

Many users keep their data in a separate database file and then create "application" databases that attach to those tables and include queries, forms, and reports that provide access to the data for specific purposes. If you've followed this design strategy, you should first convert the application databases. You can convert application databases one at a time (remember to first make a backup copy of the database file) without affecting the other databases' operation. ♦



Design Tip

Removing a text box from the tab order with the Enable and Locked properties

If you've designed many forms, you've probably needed to remove a control from the form's tab order. You've probably done so by changing the Enable property from its default setting, *Yes*, to *No*, causing Access to skip the control as you tab through the form.

Unfortunately, when you use Enable in this way to *disable* a control, Access displays the control in a weaker color than normal to indicate that you can't enter the control. For instance, if a text box has black text on a white background and you set its Enable property to *No*, Access will display light-gray text on the white background.

Sometimes, you might want to weaken the disabled control's appearance on the form. However, if you're like most users, you often want to skip a control while leaving its appearance alone. Fortunately, you

can keep the control's original, solid color by setting the Locked property as well as Enable. By default, Locked has the value *No*. You simply change it to *Yes*.

Skipping a text box

Let's look at an example. Figure A shows an order form that lets you enter product descriptions, unit prices, and the quantity purchased. The Extended Price text box then computes the extended price (quantity times price). To build this form, you create text boxes for fields named Description, Unit Price, and Quantity and an unbound text box named Extended Price. Then, assign the expression `=[Unit Price] * [Quantity]` to the Extended Price text box's Control Source property.

Since Access computes the value for Extended Price, there's no reason for the cursor to move to the text box while viewing the form. You can remove it from the tab order by changing the Enable and Locked properties. Assign *No* to Enable and *Yes* to Locked. When you view the form, you won't be able to move the cursor to the text box but the text box will still appear in the color scheme you chose. ❖

Figure A

You want the cursor to skip the Extended Price text box as you tab through the form.

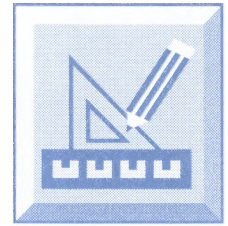
Subscribe to Inside Microsoft Access Resource Disk!

Do you wish that you could experiment with the forms, reports, tables, macros, modules, and queries we regularly feature in *Inside Microsoft Access* but don't have the time or patience to create them? If so, you may want to subscribe to Inside Microsoft Access Resource Disk. Once you subscribe, we'll send you a monthly disk loaded with all the useful tips featured in *Inside Microsoft Access*. (See the articles marked with the disk icon.)

A six-month subscription to Inside Microsoft Access Resource Disk costs \$29. A full one-year subscription is \$49. If you don't want to subscribe but would like the forms, reports, tables, macros, modules, and queries in a particular issue of *Inside Microsoft Access*, you can purchase a single disk for only \$9.95.

To subscribe or order a specific month's disk, just call Customer Relations at (800) 223-8720. Outside the US, please call (502) 491-1900.

A double-click shortcut for bringing a subform or subreport into Design View



Design Tip

When you're designing a form that has a subform, you can't actually see the subform—only the subform control. The control displays only the name of the form you're embedding. It doesn't present any information about the contents of the form.

However, you often want to refer to the form you embedded while you're developing the main

form. On these occasions, you need to bring the form into Design View so that you can examine both the main form and the embedded form.

Of course, you could bring the form into Design View by returning to the Database window, highlighting the form's name, and clicking the Design button. But Access provides a much simpler way. You can easily bring the embedded form into Design View by double-clicking the subform control. A new window will appear containing the embedded form in Design View.

For example, Figure A shows the Categories form in Design View. (You'll find this form in the NWIND.MDB sample database, located in the \ACCESS directory.) Notice that when you view the Categories form, the subform control shows only empty white space where the form named Categories Subform will appear. If you want to see Categories Subform in Design View, you can double-click anywhere in the white space. If the subform control is already selected, you'll have to select another control before double-clicking in the

Figure A

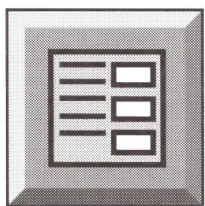
When designing the Categories form, you might want to see the subform named Categories Subform in Design View.

Figure B

To bring a subform into Design View, you simply double-click anywhere inside the subform control.

white space. Figure B shows Categories Subform in Design View.

You can use the same trick when you're designing a report that contains a subreport. When you double-click the subreport control, the report you embedded will appear in Design View in a new Design View window. ♦



Form Tip

Printing forms with white backgrounds

When you create a form by using a form wizard, the wizard paints the form's sections with a background color. Although you may like the background color while you work with the form on the screen, you won't like it when you print the form. Why not? Well, even though the background color spans the entire form, the color doesn't reach the margin. As a result, the form will print a big gray box around the controls on your form, and the margins will remain white (or the color of your paper).



Wouldn't it be nice if you could view the form with a color background but print it with a white background? In this article, we'll show you a form-design technique that does the trick.

Use full-sized rectangle controls and the Display When property

Unfortunately, Access doesn't provide a way to remove the section's background color during printing. To print the form with a white background, you must turn off the section's color permanently. However, you can display the form in color on the screen by filling every section with a rectangle control. Once you place a rectangle, you format it with the color you want and then issue the Send To Back command.

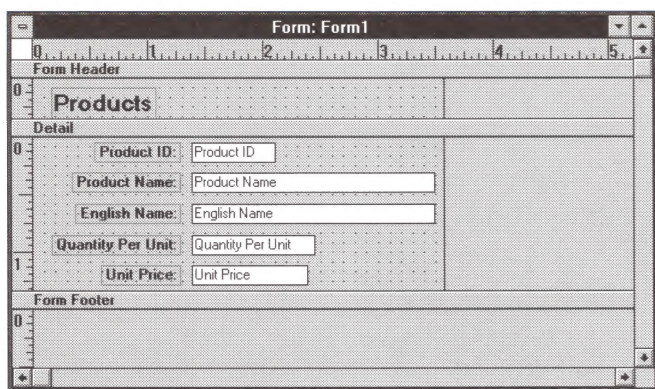
However, at this point, you've only put the color back on the form. The background color will still appear when you print the form. In order to turn off the color, you set the rectangles' Display When property to *Screen Only*. By doing so, you tell Access to show the rectangles only when you're viewing the form onscreen. Access won't display the rectangles when you print the form. Consequently, the form will print with the section's background color—white.

An example

Let's create a simple form that uses our technique. In your test database, import the NWIND.MDB database's Products table by using the File menu's Import... command. Then, highlight the Products table in the Database window and click the New Form button () on the toolbar. In the New Form dialog box, click the Form Wizards button. Select Single-Column in the following dialog box and click OK. In the next dialog box, select the fields to place on the form. For this example, choose the Product ID, Product Name, English Name, Quantity Per Unit, and Unit Price fields. Then, click the Fast-Forward button () at the bottom right of the screen. In the wizard's last dialog box, click the Design button.

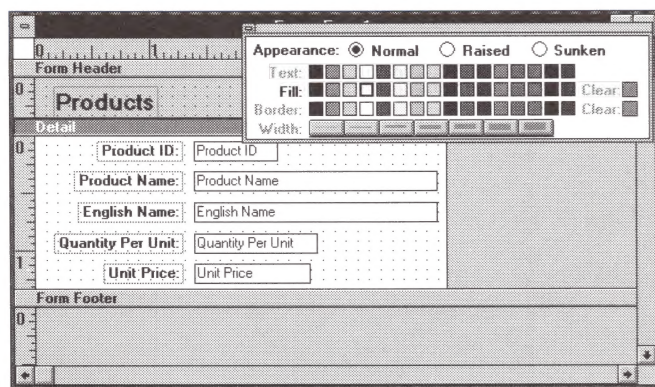
The wizard will generate the default form shown in Figure A. As you can see, the Form Header and Detail sections have a gray background. You want to view the

Figure A



The Single-Column form wizard generates a form with a gray background.

Figure B



The form looks like this when you color the Detail section white.

form with the background color but print it without that color.

Start by clicking the Palette button (■) in the tool bar so that you can assign colors to objects on the form. Next, assign white to the Detail section's background. To do so, click in an empty area of the Detail section and then, in the Palette, click the white square in the Fill row. The Detail section will turn white, as shown in Figure B.

Next, click on the rectangle tool (■) in the tool box. Position the crosshair cursor at one corner of the Detail section and drag to the diagonally opposite corner to place a rectangle control that completely fills the section. Then, move to the Palette and click the gray square in the Fill row. Next, issue the Layout menu's Send To Back command so you can see the fields again. Your form should look like the one in Figure C.

Your modified form looks exactly like the original form. The gray rectangle will look like the section's background color on the original form in both Form View and Print Preview. So that Access won't print the color rectangle, you set the rectangle's Display When property. The white background will then appear on the printed form.

If you've switched to Form View to check your form's color, click the Design View button (■) in the tool bar to return to Design View. Then, select the background rectangle. Next, open the property sheet by clicking the Properties button (■) in the tool bar. In the Display When property, enter *Screen Only*. Alternatively, you can click the property's dropdown arrow and choose *Screen Only* from the selection list.

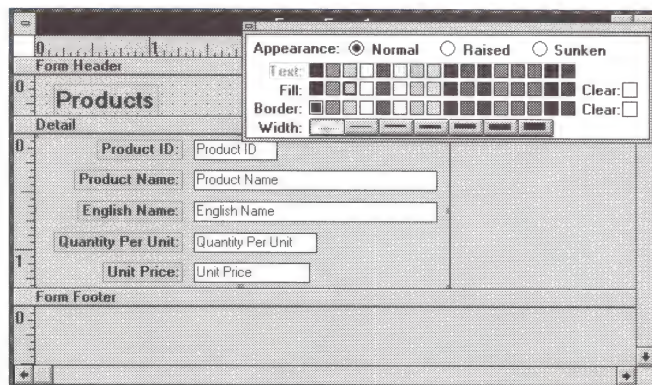
The form will still appear gray when you view it. However, when you print the form, the background will be white. Figure D shows the form in both Form View and when printed out.

Other design tips

When you want to implement this technique for new forms, you should place the rectangle after finalizing the form's layout. If you place the rectangle first, you'll find it more difficult to select and move controls than when the rectangle isn't present.

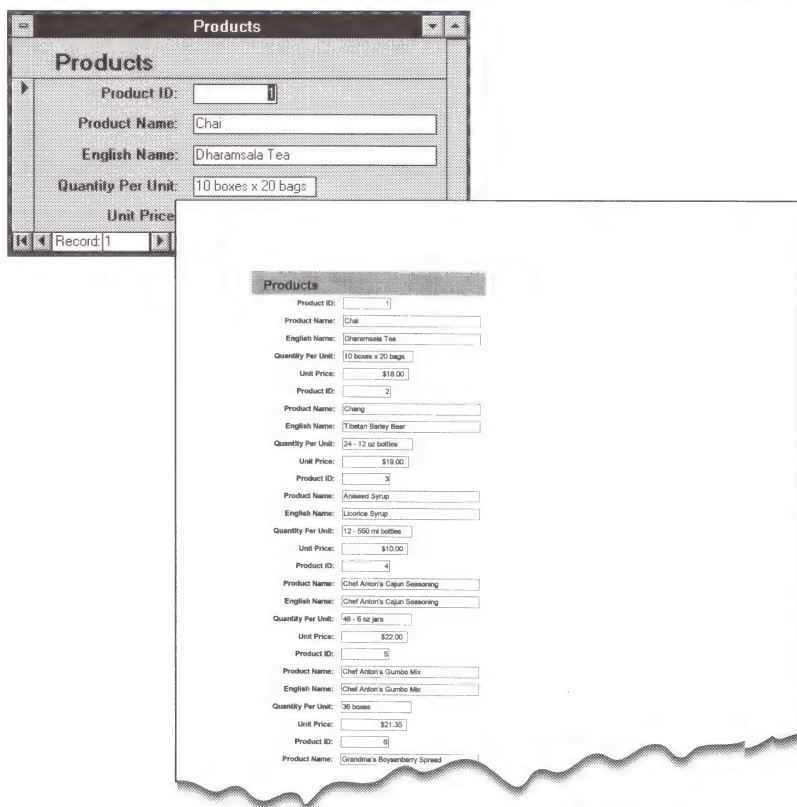
You can also set a command button's Display When property to *Screen Only*.

Figure C



Color the Detail section by placing a gray rectangle in the background.

Figure D



When you print the form, the gray background disappears.

Obviously, buttons are useless on the printed page.

Conclusion

In this article, we showed you how to create a background color that disappears when you print the form. Instead of coloring the form sections, you place a rectangle that displays your background color only while you view it onscreen. ♦



Printing addresses directly onto envelopes

The Mailing Label wizard provides an easy way to format address data for printing on mailing labels. However, you don't need to generate mailing labels to address envelopes. Instead, you can design a report that prints the address information as it would appear on the envelope. You then print the report and feed envelopes directly into your printer. In this article, we'll show you how to set up the report for printing directly onto envelopes.

Configuring the paper size for envelopes

To print onto envelopes, you must set up the report to print on the envelope paper size. To do so, choose Print Setup... from

the File Menu to open the Print Setup dialog box. Depending on the capabilities of your printer, you can set up the report for printing onto envelopes by using simple settings in this dialog box. You'll probably find envelope settings in both the Size and Source combo boxes of the Paper section of the Print Setup dialog box. You may also need to click the Landscape radio button in the Orientation section to print the address information along the length of the envelope. The precise settings will vary according to the type of printer you have.

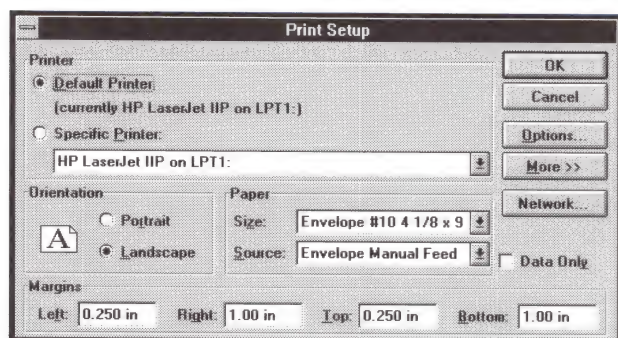
You should also move to the Margins section and enter 0.25 as all four margin settings so that the information prints as close to the corners as possible. For instance, the return address belongs in the upper-left corner of the envelope.

Let's look at an example. When we created a report for the Hewlett-Packard LaserJet IIP, we discovered that the Paper section's combo boxes offered envelope-specific settings. We selected the entries *Envelope #10 4 1/8 x 9 1/2* from the Size combo box's selection list and *Envelope Manual Feed* from the Source combo box's selection list. We also clicked the Landscape radio button to properly orient the address data for printing onto envelopes. After we made these selections, our Print Setup dialog box looked like the one shown in Figure A.

Once you've configured the Print Setup dialog box, you can return to the report's Design View and place the address information as appropriate. Figure B shows our sample envelope report. We used label controls to place the return address, since that information doesn't change. Also, we used text boxes to place the mailing address in the middle of the envelope.

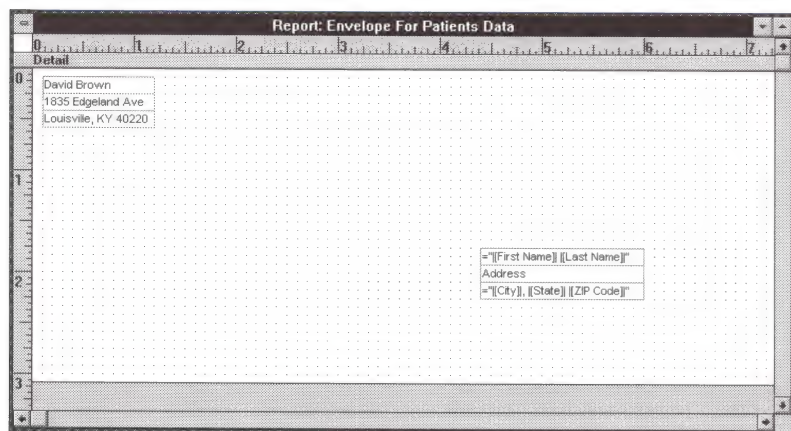
Note that we used expressions rather than fields as the control source of the first and last lines of the address. The expression concatenates fields so they print as one entry. If you're unfamiliar with our use of the vertical bars in the expressions, refer to "Combining Text with Field Entries in Form and Report Controls" in the October 1993 issue. ♦

Figure A



After you set up the report for printing on envelopes, the Print Setup dialog box should look like this.

Figure B



After you set up the report for printing on envelopes, you place the address.

Defining a combo box's selection list based on another control in the form

I would like to define a combo box's selection list based on another field in the form. For instance, in the database I'm setting up for an automotive business, I use one field named Car Manufacturer and another named Car Model. I want to create a Car Model combo box that limits its entries to the models produced by the manufacturer displayed in the Car Manufacturer text box. For example, if I enter *Ford* in the Car Manufacturer text box, I want the Car Model combo box to list only Ford vehicles.

I would also like to know a way to insert information in a field based on an entry of another field. For instance, if I enter a ZIP code, I want the form to enter the city and state.

*Miles Wada
Torrance, California*

The form-design techniques you'd use to implement Mr. Wada's two requests are essentially the same. In both cases, the form contains a control whose value depends on the entry in another control. In general terms, you create a macro that satisfies the dependency between the two fields. You then assign the macro to an appropriate event property so Access will run the macro whenever it must enforce the relationship between the two controls.

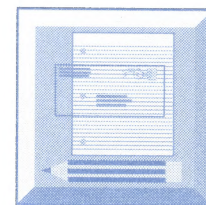
Let's start by addressing Mr. Wada's second request. He wants to fill in City and State text boxes based on the ZIP code entry.

You can pull this off only if you have a table that provides the city and state for every ZIP code you might enter. If you have such a table, you first create a macro that assigns the appropriate city and state values from the table to the city and state controls on the data entry form. You then assign the macro to the ZIP code text box's After Update property. That way, Access will fill in the City and State text boxes after you enter a ZIP code. (In creating the macro, you call the DLookup() function to find the city and state values for the ZIP code and use the SetValue action to assign the field entries.)

A step-by-step example

Now that we've outlined the basic technique, let's work through Mr. Wada's primary request—creating a combo box that provides entries in the selection list based on the value of another control. We'll create an entry form with a combo box that provides allowable Car Model values and a query that selects those models from the value you've entered in the Car Manufacturer control.

We'll first define the tables we'll use in the example. The Repair Orders table will store information about a customer's repair order—including the manufacturer and model of the car. The Car Models table will store information about the available models and their manufacturers. Figure A shows the structure of these tables and some sample data.



Letters

Figure A

Table: Repair Orders						
Repair Order ID	Customer Name	Customer Phone	Car Manufacturer	Car Model	Estimate Cost	Actual Cost
1	Fred Jones	555-2342	Ford	Bronco II	\$345.00	\$360.00
2	Mike Jones	555-8792	GM	Blazer	\$240.00	\$250.00
3	James Jones	555-2764	Chrysler	Jeep Cherokee	\$700.00	\$725.00
4	Ida Jones	555-2299	Toyota	Land Cruiser	\$800.00	\$700.00
* (Counter)					\$0.00	\$0.00

Table: Car Models		
Model ID	Model Name	Manufacturer
1	Bronco II	Ford
2	Blazer	GM
3	Jeep Cherokee	Chrysler
4	PathFinder	Nissan
5	Land Cruiser	Toyota
6	4 Runner	Toyota
7	S-10 Pickup	GM
8	Suburban	GM
9	Explorer	Ford
* (Counter)		

We'll use the Repair Orders and Car Models tables in our example.

After you create these tables, develop a Repair Order Entry form that offers a combo box to list a manufacturer's models. Use the Single-Column form wizard to create the default form for the Repair Orders table. Then, save the form by pulling down the File menu and selecting the Save As... option. In the Save As dialog box, enter *Repair Orders Entry* and click OK.

Next, replace the Car Model text box with a combo box. Open the property sheet by clicking the Properties button (🔧) on the tool bar, and assign *Car Model Combo* to the Control Name property. You should also put the new combo box in its proper

spot in the form's tab order. To do so, issue the Edit menu's Tab Order... command and, in the Tab Order dialog box, click the Auto Order button; then, click OK.

Before setting the other combo box properties, you must create the query the combo box will use as the row source. Return to the Database window, highlight the Car Models table, and click the New Query button (🔍) in the tool bar. Then, drag the Model Name and Manufacturer fields to the QBE grid. In the Manufacturer column, deselect the Show check box and enter the identifier

```
[Forms]![Repair Orders Entry]!
[Car Manufacturer]
```

in the Criteria cell. This identifier accesses the value you entered in the Car Manufacturer text box. Figure B shows the final query. Pull down the File menu's Save As... option, enter *Models From A Manufacturer*, and click OK. Then, close the query window.

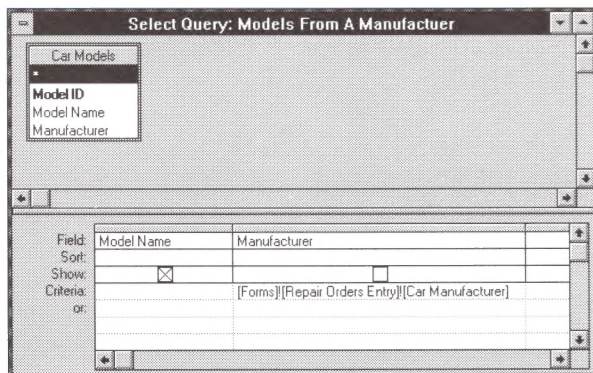
Now return to the Repair Orders Entry form. You should have left the form in Design View so that you can move directly to the Car Model Combo control's property

sheet. Then, assign *Models From A Manufacturer* to the Row Source property. That way, Access will display the records the query selects in the combo box's list. You should also enter *Yes* in the Limit To List property field so that Access won't let you enter invalid car models.

You've almost finished. Access will not rerun the Models From A Manufacturer query when you change the Car Manufacturer text box. You must create a macro that forces a requery and then assign the macro to the combo box's On Enter property. By using On Enter to trigger the requery, you'll ensure that the combo box's selection list will always be in sync with the current Car Manufacturer entry.

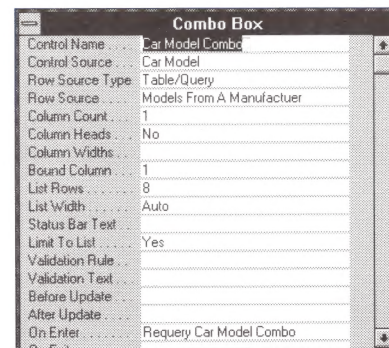
To create the macro, return to the Database window, click the Macro button, and then click the New button. In the first row, enter *Requery* in the Action cell. Then, in the Control action argument, enter *Car Model Combo*. Save the macro with the name *Requery Car Model Combo*. Finally, return to the form, open the Car Model Combo control's property sheet, and assign *Requery Car Model Combo* to the On Enter property. Figure C shows the final property sheet.

Figure B



You create this query to select all models from the manufacturer listed in the Car Manufacturer text box.

Figure C



After you assign the macro to the On Enter property, your property sheet will look like this.

The ANALYZER.MDA utility lets you print out table specs and more

Is there some way to print out table specs? I'm working with a vendor on a joint database, and I have no way of listing field names, sizes and data types, key fields, defaults, and so on. I could copy the

information by hand, but that could take hours for a table with a lot of fields.

Kurt Springman
Bellevue, Washington

If you upgraded to Access 1.1, you already have a utility that'll do the job: The ANALYZER.MDA database is a full-blown application that examines your database's tables and stores all the table and field information in a new table. You can then create reports or forms to review the results.

Furthermore, the ANALYZER.MDA database doesn't itemize just tables—it can also analyze your queries, forms, reports, and macros. In this article, we'll show you how to take advantage of this incredibly useful utility with your databases. The process for using the analyzer may seem complicated, but it's certainly worth the effort.

Before launching Access, you must open the Notepad application and modify the MSACCESS.INI file—Access' initialization file. Double-click Notepad's icon () in the Accessories group. Then, pull down Notepad's File menu and select the Open... command. The default directory should be the Windows directory, which is where the initialization file resides. If not, select the Windows directory in the Directories area. Then, type *MSACCESS.INI* in the File Name text box and click OK. When the file appears, scroll down until you find a section marked with the header *[Libraries]*.

The Libraries section should already contain at least one entry—*wizard.mda=ro*. Create a new line after the existing one and type

```
analyzer.mda=
```

Next, issue the Save command from the File menu and close Notepad. You've just defined the ANALYZER.MDA database file as a library database, which means you can run the Analyzer utility's Access Basic functions from the current database.

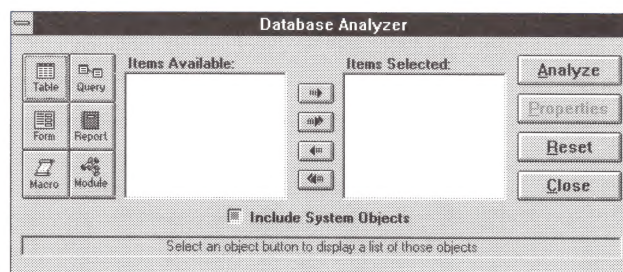
Next, launch Access and open the database you want to analyze. For the purposes of this example, open the PIM.MDB database, which resides in the \ACCESS directory. Be sure to hold down the [Shift] key as you select the database file from the Open Database dialog box, as PIM.MDB has an Autoexec macro.

You then create a macro that launches the Analyzer utility. Click the Macro button to list the database's macros and

then click the New button. In the first row's Action cell, enter *RunCode*. Then, press [F6] and type *StartAnalyzer()* in the Function argument. Finally, save the macro by using the File menu's Save As... command. Enter *StartAnalyzer* in the Save As dialog box and click OK.

Finally, run the new StartAnalyzer macro by clicking the Run button () in the tool bar. The Database Analyzer window shown in Figure A will appear. You use this window to choose the database objects you want to investigate.

Figure A



You use the Database Analyzer window to choose the database objects you want to investigate.

You'll notice the familiar object buttons from the Database window on the left side of the Database Analyzer window. When you click a button, the objects of that type will appear in the Items Available list box. For instance, if you click the Table button, the PIM database's tables will appear.

Then, select the objects for which you want to gather information. You select objects the same way you select fields in a form or report wizard—by either double-clicking the object names in Items Available or highlighting the names and clicking the arrow buttons. When you select an object, it appears in the Items Selected list box.

The items you select appear with their object type as a tag. Consequently, the Items Selected list box can display all the objects of all object types. For example, if you select the Address table, the entry *Table: Address* will appear in the Items Selected list box. You can then click the Query button and select the AddressQry object. The entry *Query: AddressQry* will appear below *Table: Address*.

After you've selected all the objects you want to analyze, you click the Analyze

Microsoft Access
Technical Support
 (206) 635-7050

Please include account number from label with any correspondence.

button. The Analyzer will then present the Select An Output Database dialog box. You specify the database in which you want the Analyzer to store the results. The Analyzer then creates several tables in that database to store the results. Table A lists the tables with brief descriptions of the results the Analyzer puts in them.

You must provide an existing database file in the Select An Output Database dialog box. If you want to store the results in a new database, you must create the database before running the Analyzer. However, you should consider specifying the database you're analyzing. You won't

confuse the Analyzer's tables with your database's tables, because the Analyzer tables begin with the @ symbol. As a result, the tables will appear in a group at the top of your list of tables. They will always be present as you design new objects in your database, so you can refer to them at any time. Furthermore, you can easily delete them when you no longer need them.

Notes

Access 2.0 has replaced the Analyzer utility in Access 1.1 with the Database Docu-
 menter add-in. We'll describe this new utility in a future article. ♦

Table A

Table Name	Description	Table Name	Description
@FormControls	Lists all the controls on every form you selected; provides the values of several control properties	@QueryDetails	Lists data on the fields of every query you selected; returns the field names, the data types and field sizes, and the source table but does not return query criteria
@FormProperties	Lists all the forms you selected; provides the values of several form properties	@ReportControls	Lists all the controls on every report you selected; provides the values of several control properties
@MacroDetails	Lists the actions of every macro you selected	@ReportProperties	Lists all the reports you selected; provides the values of several report properties
@ModuleProcedures	Lists the procedure names and their arguments for every module you selected	@TableDetails	Lists field information for every table you selected; returns the field names, data types, and field size
@ModuleVariables	Lists the variables and their data types in all functions		
@QuerySQL	Lists the equivalent SQL statement for every query you selected		

